

Рисунок 3 – Регулируемое крепление

Использование регулируемых антенных опор на периметре объектов позволит производить по месту точную настройку симметрии емкости антенного полотна. При точной настройке емкостного полотна, исключается применение «подстроечного» конденсатора, тем самым повышается стабильность работы емкостных датчиков обнаружения. При использовании регулируемых креплений, замена изоляторов непригодных к использованию производится с наименьшими трудозатратами.

ЛИТЕРАТУРА

1. "Безопасность Окружающей Среды" №3-2007: Безопасность ядерных и радиационных объектов. Севрюков Д.В., Асфандияров А.Х. , – М.: Вымпел. – 2012, – 121с
2. <http://vi.elsys.ru/storage/nto.pdf>
3. Магауенов Р.Г. Системы охранной сигнализации: основы теории и принципы построения: Учебное пособие. – М.: Горячая линия. – Телеком, 2007- 367с.

УДК 004.272.44

Р.А. Шуленков, магистрант
(БГТУ, г. Минск)

ТАКТИРОВАНИЕ, СИНХРОНИЗАЦИЯ И ФАПЧ ПРОЦЕССОРА

Тактирование и синхронизация - это основные элементы работы любого процессора (микроконтроллера и микропроцессора). Работа тактирования начинается с работы одного из следующих компонентов внутреннего или внешнего калибруемого РС-генератора, внешнего кварцевого резонатора или внешнего керамического резонатора. Тактирование: с сопровождением данных сигналами тактирования, с выработкой сигналов тактирования приемников. Проблема расфазирования тактовых импульсов для быстродействующих ЦПУ (центральное процессорное устройство) актуальна, что на современные БИС/СБИС (большие интегральные схемы/сверх большие интегральные схемы) устанавливают специальные модули, улучшающие синфазность тактовых сигналов в различных областях схем и при необходимости выполняющими и преобразования частоты (умножение, деление).

Для фазирования (коррекции) тактовых импульсов используют PLL (Phase Locked Loop), FLL (Frequency Locked Loop), DLL (Delay Locked Loop), DCM (Digital Clock Manager). Наибольшее распространение получила система фазовой автоподстройки частоты (ФАПЧ, PLL). Следует отметить, что PLL, по крайней мере, в реализуемых сейчас вариантах, - аналоговые устройства. Действительно, компенсируемый сдвиг фазы синхросигналов может иметь любое значение (является непрерывной величиной), поэтому и система его точной отработки должна работать в аналоговом режиме. С этой точки зрения PLL трудны для реализации. Являясь замкнутыми нелинейными системами, PLL могут оказаться неустойчивыми, что требует их для обеспечения надежной устойчивой работы во всем диапазоне изменения температуры и других воздействий[1].

Системы фазовой автоподстройки частоты представляют собой схемы с отрицательной обратной связью, которые состоят из генератора, управляемого напряжением (VCO) и фазового компаратора, которые включаются таким образом, что генератор поддерживает постоянный угол фазы по отношению к опорному сигналу. ФАПЧ может использоваться, например, для создания выходного сигнала со стабильной частотой из низкочастотного сигнала фиксированной частоты[2].

В ФАПЧ сигнал рассогласования от фазового компаратора пропорционален разности фаз между сигналом на входе и сигналом в цепи обратной связи. Среднее значение сигнала на выходе фазового детектора будет оставаться постоянным в том случае, если входной сигнал в цепи обратной связи будет иметь одинаковую частоту.

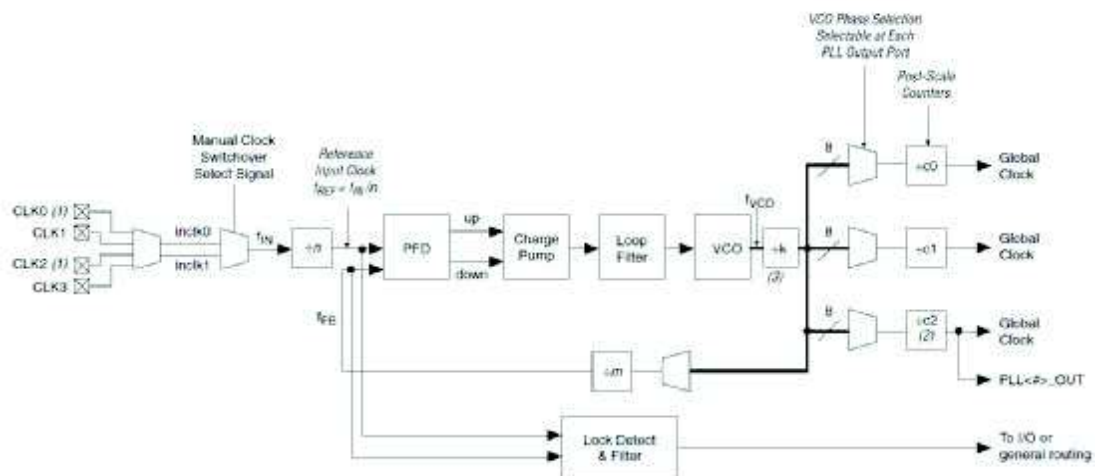


Рисунок 1 – Базовая PLL архитектура (Altera Cyclone II)

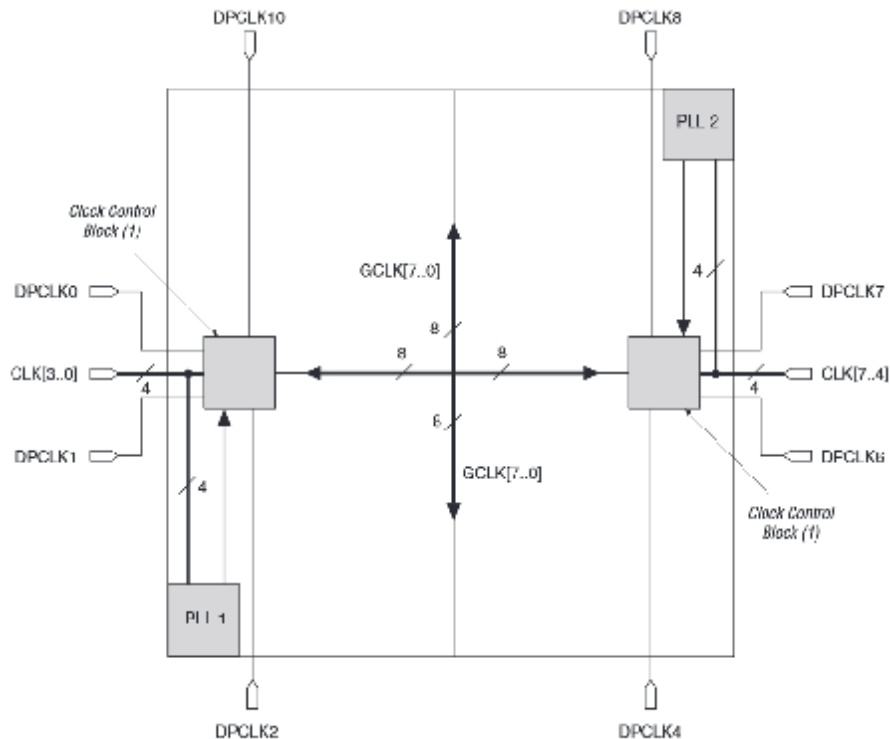


Рисунок 2 – Блок управления частотой (Altera Cyclone II)

Рассмотрим построение системы тактирования процессора на базе FPGA (Field-Programmable Gate Array - программируемая пользователем вентильная матрица) Altera Cyclone II EP2C5T144. Характеристика тактирования, данного FPGA: 2 области PLLs, 8 CLK выводов, 20 DPCLK выводов, 16 глобальных линий тактирования[3].

Каждый микроконтроллер имеет свою уникальную структуру внутреннего тактирования. Например, система тактирования STM32F756NGHx (ARM Cortex M7) имеет отдельные линии тактирования для LCD-TFT, USB и I2S. Внутренняя структура наполнения большим количеством делителей и умножителей, последнее является характерной чертой для АНВ (Advanced High-performance Bus) – АРВ (Advanced Peripheral Bus) моста. АНВ – АРВ – мост между шинами АНВ и АРВ; используется для подключения различных АРВ – устройств, таких как компоненты отладки, к шине встроенных периферийных устройств процессора ARM. Кроме того, конструкция процессора позволяет производителям микросхем подключать дополнительные АРВ – устройства к внешней шине встроенных периферийных устройств (PPB - Private Peripheral Bus) с помощью этой шины[4].

Формирование спецификации системы тактирования разрабатывается на языках проектирования цифровой аппаратуры VHDL, Verilog, SystemVerilog. Далее создается функциональное описание работы системы тактирования для будущей микросхемы и происходит верификация. Современные микросхемы часто называются системами

на кристалле (SoC) и содержат множество различных блоков, работающих параллельно и взаимодействующих между собой. Это сложное устройство, которое требует детальной проверки на правильность функционирования и отсутствие ошибок в проектном описании. Под проверкой подразумевается разработка детального плана и среды верификации на языке SystemVerilog с использованием методологии UVM (Universal Verification Methodology), её отладка, написание тестов, создание программной модели по спецификации, подготовка скриптов компиляции и запуска среды, скриптов анализа результатов множественных запусков. Верификация проекта цифровой интегральной схемы (ИС) — один из самых затратных по времени этапов разработки. Для достижения достаточной полноты проверки необходимо провести моделирование на большом наборе тестов и, если выявлены ошибки, после корректировки необходимо весь процесс моделирования повторить.

Формирование системы тактирования для архитектуры процессора произведем, используя Quartus II для FPGA и CPLD (complex Programmable Logic Device). Блок Altera Phase-Locked Loop (ALTPLL) использует внутренний PLL в связке с умножителями и делителями.

На вход PLL подается входная частота опорного генератора F_{in} . Далее на счетчике n опорная частота делится и получается иная частота F_{ref} , которая поступает на фазовый детектор PFD (phase-frequency detector). Фазовый детектор сравнивает фазы частот F_{ref} и той, что поступает с делителя m . Разность фаз, фильтруется и управляет генератором VCO. На выходе управляемого генератора - новая частота F_{vco} . Фазовый детектор подает управляющее воздействие на генератор VCO до тех пор, пока не выполнится условие $F_{ref} = F_{vco}/m$. При этом условия частоты, подаваемые на детектор фаз равны. Таким образом, например, если $m=2$, то частота генератора VCO должна получиться в 2 раза выше частоты F_{ref} . Последний этап - частота F_{vco} делится на выходном счетчике k . Подбирая коэффициенты n, m, k можно синтезировать довольно большой диапазон частот. Более сложные компоненты PLL позволяют перезагружать коэффициенты в процессе работы системы.

Блок Altera Phase-Locked Loop

<pre> LIBRARY ieee; USE ieee.std_logic_1164.all; LIBRARY altera_mf; USE altera_mf.all; ENTITY PLL IS PORT (inclk0 : IN STD_LOGIC := '0'; </pre>	<pre> BEGIN sub_wire4_bv(0 DOWNTO 0) <= "0"; sub_wire4 <= To_stdlogicvector(sub_wire4_bv); sub_wire1 <= sub_wire0(0); c0 <= sub_wire1; sub_wire2 <= inclk0; sub_wire3 <= sub_wire4(0) </pre>
---	--

<pre> c0 : OUT STD_LOGIC); END PLL; ARCHITECTURE SYN OF pll IS SIGNAL sub_wire0 : STD_LOGIC_VECTOR (5 DOWNT0 0); SIGNAL sub_wire1 : STD_LOGIC ; SIGNAL sub_wire2 : STD_LOGIC ; SIGNAL sub_wire3 : STD_LOGIC_VECTOR (1 DOWNT0 0); SIGNAL sub_wire4_bv : BIT_VECTOR (0 DOWNT0 0); SIGNAL sub_wire4 : STD_LOGIC_VECTOR (0 DOWNT0 0); COMPONENT altpll GENERIC (clk0_divide_by : NATURAL; clk0_duty_cycle : NATURAL; clk0_multiply_by : NATURAL; clk0_phase_shift : STRING; compensate_clock : STRING; inclk0_input_frequency : NATURAL; intended_device_family : STRING; lpm_hint : STRING; lpm_type : STRING; operation_mode : STRING; port_activeclock : STRING; port_areset : STRING; port_clkbad0 : STRING; port_clkbad1 : STRING; port_clkloss : STRING; port_clkswitch : </pre>	<pre> DOWNT0 0) & sub_wire2; altpll_component : altpll GENERIC MAP (clk0_divide_by => 3, clk0_duty_cycle => 50, clk0_multiply_by => 2, clk0_phase_shift => "0", compensate_clock => "CLK0", inclk0_input_frequency => 20000, intend- ed_device_family => "Cyclone II", lpm_hint => "CBX_MODULE_PREFIX=PLL", lpm_type => "altpll", operation_mode => "NORMAL", port_activeclock => "PORT_UNUSED", port_areset => "PORT_UNUSED", port_clkbad0 => "PORT_UNUSED", port_clkbad1 => "PORT_UNUSED", port_clkloss => "PORT_UNUSED", port_clkswitch => "PORT_UNUSED", port_configupdate => "PORT_UNUSED", port_fbin => "PORT_UNUSED", port_inclk0 => "PORT_USED", port_inclk1 => "PORT_UNUSED", port_locked => "PORT_UNUSED", port_pfdena => "PORT_UNUSED", port_phasecounterselect => "PORT_UNUSED", port_phasedone => "PORT_UNUSED", </pre>
--	---

STRING;			port_phasestep	=>
	port_configupdate	:	"PORT_UNUSED",	
STRING;			port_phaseupdown	=>
	port_fbin	:	"PORT_UNUSED",	
STRING;			port_pllana	=>
	port_inclk0	:	"PORT_UNUSED",	
STRING;			port_scanaclr	=>
	port_inclk1	:	"PORT_UNUSED",	
STRING;			port_scanclk	=>
	port_locked	:	"PORT_UNUSED",	
STRING;			port_scanclkena	=>
	port_pfdana	:	"PORT_UNUSED",	
STRING;			port_scandata	=>
	port_phasecounterselect		"PORT_UNUSED",	
	: STRING;		port_scandataout	=>
	port_phasedone	:	"PORT_UNUSED",	
STRING;			port_scandone	=>
	port_phasestep	:	"PORT_UNUSED",	
STRING;			port_scanread	=>
	port_phaseupdown	:	"PORT_UNUSED",	
STRING;			port_scanwrite	=>
	port_pllana	:	"PORT_UNUSED",	
STRING;			port_clk0	=>
	port_scanaclr	:	"PORT_USED",	
STRING;			port_clk1	=>
	port_scanclk	:	"PORT_UNUSED",	
STRING;			port_clk2	=>
	port_scanclkena	:	"PORT_UNUSED",	
STRING;			port_clk3	=>
	port_scandata	:	"PORT_UNUSED",	
STRING;			port_clk4	=>
	port_scandataout	:	"PORT_UNUSED",	
STRING;			port_clk5	=>
	port_scandone	:	"PORT_UNUSED",	
STRING;			port_clkkena0	=>
	port_scanread	:	"PORT_UNUSED",	
STRING;			port_clkkena1	=>
	port_scanwrite	:	"PORT_UNUSED",	
STRING;			port_clkkena2	=>
	port_clk0	:	"PORT_UNUSED",	
STRING;			port_clkkena3	=>
	port_clk1	:	"PORT_UNUSED",	
STRING;			port_clkkena4	=>
	port_clk2	:	"PORT_UNUSED",	
STRING;			port_clkkena5	=>
	port_clk3	:	"PORT_UNUSED",	
STRING;			port_extclk0	=>
	port_clk4	:	"PORT_UNUSED",	

STRING;	port_clk5	:	port_extclk1	=>
STRING;	port_clkena0	:	"PORT_UNUSED",	
STRING;	port_clkena1	:	port_extclk2	=>
STRING;	port_clkena2	:	"PORT_UNUSED",	
STRING;	port_clkena3	:	port_extclk3	=>
STRING;	port_clkena4	:	"PORT_UNUSED"	
STRING;	port_clkena5	:)	
STRING;	port_extclk0	:	PORT MAP (
STRING;	port_extclk1	:	inclk => sub_wire3,	
STRING;	port_extclk2	:	clk => sub_wire0	
STRING;	port_extclk3	:);	
STRING			END SYN;	
);				
PORT (
clk	:	OUT		
STD_LOGIC_VECTOR (5 DOWNT0 0);				
inclk	:	IN		
STD_LOGIC_VECTOR (1 DOWNT0 0)				
);				
END COMPONENT;				

Созданный блок тактирования является основой работы любой архитектуры процессора, дальнейшее формирование частоты реализуется для работы внутренних шин процессора и внутренней периферии (UART/USART, SPI, I2C, SDIO и др.).

ЛИТЕРАТУРА

1. Цифровая схемотехника, Евгений Угрюмов, 2010
2. Линейные схемы. Руководство по проектированию, 2011
3. Старт ПЛИС-ы. Первая ласточка (mcsu.by)
4. Ядро Cortex-M3 компании ARM, Ю. Джозеф, 2015